Cybernetics and Systems Analysis, Vol. 43, No. 5, 2007

NEW MEANS OF CYBERNETICS, INFORMATICS, COMPUTER ENGINEERING, AND SYSTEMS ANALYSIS

ARCHITECTURAL OPTIMIZATION OF A DIGITAL OPTICAL MULTIPLIER

A. V. Anisimov and I. A. Zavadskyi

UDC 681.325.57

A computational model of estimation of the time complexity of logical circuits constructed from elements of an optical element base is investigated. A fast parallel multiplier is constructed.

Keywords: *optical switch, switching element, multiplier, logical circuit, synchronous arithmetic.*

In the past decade, the search for new physical principles that could underlie future processors is activated in designing high-speed parallel computers. Increasing attention is being given to light as an information carrier. On the one hand, the development of nanophotonic technologies in the near future will make it possible to control individual photons as bits in a quantum computer. On the other hand, the control of ultrashort laser pulses makes it possible to perform classical computations at tens of gigahertz. A change in the basic physics of computers requires new approaches to the modeling of their functioning. In Sec. 1 of this article, a model of computations is investigated that allows one to estimate the speed of logical circuits with allowance made for the distinctive features of processing optical signals. In Sec. 2, this model is used in designing a parallel optical multiplier and, in Sec. 3, its time and space complexity are estimated and compared with well-known types of parallel multipliers, in particular, with the device proposed in [1].

A reason for investigations in this direction became the results of current works oriented toward the creation and perfection of miniature optical switches in many research centers of Japan, Western Europe, and the USA. Several companies produce industrial models of such devices. Their characteristics allow one to create networks consisting of several switches or several tens of sequentially connected switches without any additional equipment such as optical amplifiers. This is amply sufficient to widely use optical switches in telecommunications but insufficient for construction of high-speed logical circuits and low-power computers. Nevertheless, the creation of optical switches that can be used as an element base of universal processors is the objective of several international programs. Taking into account the high rate of improving the characteristics of optical switching elements during the past 5–7 years, the creation of an efficient digital optical processor can be considered as a problem that can be solved in the nearest decade.

It may be noted that the most promising basic elements of optical logical circuits are Fabry– Perot microresonators and Mach-Zehnder switches in photon crystals. A detailed description of the principle of action and characteristics of these devices can be found in [2] and [3], respectively, and a brief review of them is given in [1].

1. MODELING OF OPTICAL COMPUTERS

In contrast to the majority of investigations in the field of optical computations, the authors focus their attention not on the optimization of physical parameters of an element base but on the construction of logical circuits in which the distinctive features of optical switches are used with maximal efficiency. We consider erroneous the widespread approach to the creation of optical logical circuits that consists of the construction of switches from some collection of basic gates, for example, disjunctions, conjunctions, negations, one-digit adders, etc. with successive replacement of the corresponding elements in well-known circuits constructed from the traditional transistor-resistor element base by these gates. The

Taras Shevchenko University, Kiev, Ukraine, *mi@unicyb.kiev.ua; zava@ukr.net.* Translated from Kibernetika i Sistemnyi Analiz, No. 5, pp. 165–177, September–October 2007. Original article submitted May 14, 2007.

1060-0396/07/4305-0749

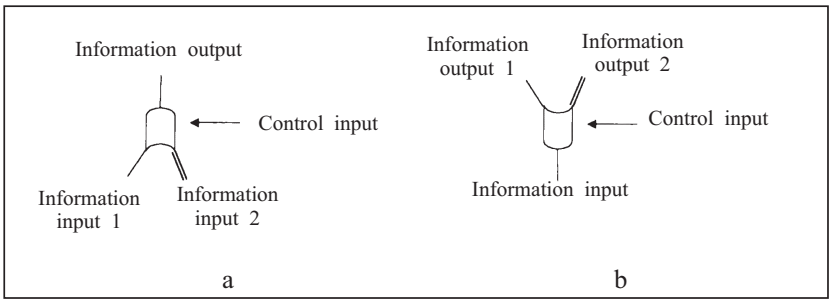

Fig. 1. Switching element: (a) with two information inputs and one output; (b) with one information input and two outputs.

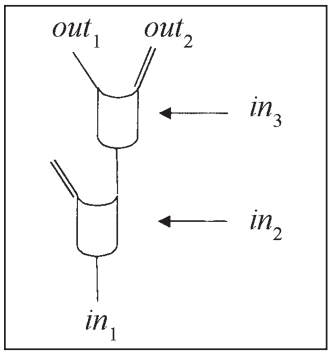

Fig. 2. A single-clock switch circuit.

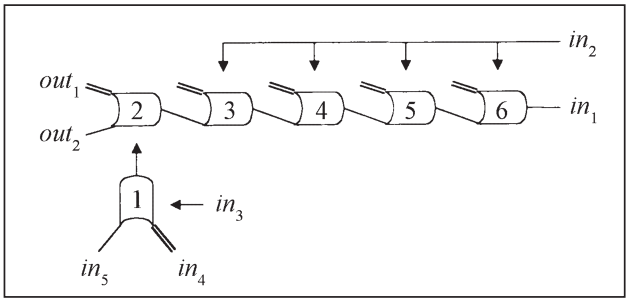

Fig. 3. Switching circuit whose time complexity depends on the value of t_{sw} / t_{trans}.

inefficiency of this approach is conditioned by the fact that the model of computations that is used for the estimation of the complexity of traditional logical circuits does not allow one to adequately estimate time characteristics of optical circuits. To prove this thesis, let us consider some key distinctive features of digital optical computations.

All the signals in computing circuits constructed from electronic logical elements possess equal rights but, in optical switches, one must distinguish between two types of signals, namely, control and information ones. These signals are usually of different nature, for example, a control signal can be electric and an information signal can be optical or both signals can be light fluxes with different wavelengths. The interaction between control and information signals is underlain by a definite physical process, for example, the pumping or relaxation of a resonator. The speed of running of this process and also the speed of transformation of an information signal into a control one determine the switch time of an optical element (we denote it by t_{sw}). The time of transfer of an information signal through an elementary optical device (t_{trans}) is almost the same as the time of its transmission along a fragment of a waveguide that has the same length as the switch itself. The value of t_{trans} must also take into account the time of passage of the minimally possible distance between switches by the information signal. For the majority of modern optical switches, the ratio t_{sw} / t_{trans} is within the range from several unities to several hundreds. Thus, there is a potential possibility to accelerate computations by constructing logical circuits so that a minimal possible number of control signals are sequentially computed.

An abstraction of a switch is a switching element (SE). It has no more than two information inputs and outputs and also one control input. Depending on the value arriving at the control input, signals from some information inputs are transmitted to definite information outputs. In [1] and [4], SEs with two information inputs and one information output (Fig. 1a) were considered. For the unit control signal, the information output is connected with the input denoted by a single line and, for the zero one, it is connected with the input denoted by a double line. In this case, the signal at the information input that is not connected with the output is lost. Such losses can be avoided with the help of an SE with one information input and two outputs (Fig. 1b). For the zero control signal, the input is connected with the output denoted by a double line and, for the unit signal, it is connected with the output denoted by a single line. In this case, the signal is absent at the output that is not connected to the input, i.e., this output is considered to be zero. In what follows, we will use precisely such SEs that do not lead to losses of information signals and, hence, allow one to construct circuits with considerably low-power consumption. It is precisely one SE with one information input and two outputs that models a Mach–Zehnder switch.

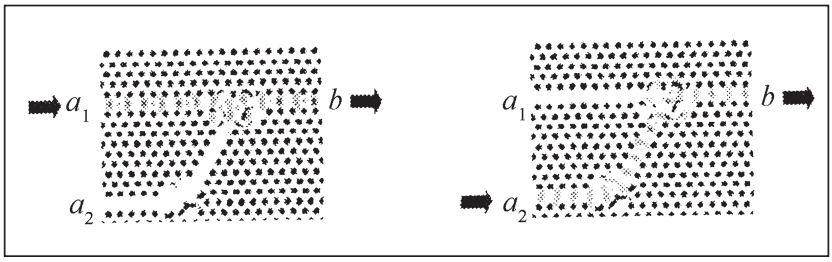

Fig. 4. *Y*-connection of waveguides that realizes the operation OR in a photon crystal.

If we connect information inputs of some SEs to information outputs of others SEs, then a circuit will be constructed in which all the SEs are simultaneously switched, i.e., a single-clock switch circuit. The total operation time of such a circuit equals $t_{sw} + Lt_{trans}$, where L is the number of SEs in the longest path connecting an arbitrary input with an arbitrary output. For example, the operation time of the circuit presented in Fig. 2 equals $t_{sw} + 2t_{trans}$. This principle of determination of time complexity is easily generalized to the case when a computer consists of several single-clock circuits connected so that any of them can begin its operation only after the completion of operation of all the circuits on which it depends. Devices of this type that are designed for the execution of most important arithmetic operations are considered in [1]. In many cases, their time complexity determined as above is essentially less than the time complexity of devices performing the same operations with the highest speed from the viewpoint of the classical approach according to which the operation time of a circuit is determined by the largest number of elements along the path from its inputs to outputs.

However, the principle described above does not allow one to adequately estimate time complexities of switching circuits in the general case since, depending on the value of t_{sw} / t_{trans}, the operation time of the same circuits can be determined by different factors. For example, let us consider the circuit presented in Fig. 3. If we assume that signals simultaneously arrive at all the inputs of the circuit at the moment of time 0, then SE 2 switches at the moment of time $2t_{sw} + t_{trans}$ and its information input is computed at the moment $t_{sw} + 4t_{trans}$. Hence, if we have $t_{sw} > 3t_{trans}$, then the operation time of the circuit equals $2t_{sw} + 2t_{trans}$, and if we have $t_{trans} < t_{sw} \le 3t_{trans}$, then its time complexity amounts to $t_{sw} + 5t_{trans}$.

Let us consider the formula that specifies the operation time of switching circuits and allows one to solve the mentioned problems. As is easy to see, in the general case, the operation time of a circuit s is determined by the value of

$$\max_{v \in V_s} \{sw(v)t_{sw} + l(v)t_{trans}\}. \qquad (1)$$

Here, V_s is the set of all the paths from the inputs of the circuit s to its outputs, v is some path, i.e., a sequence of adjacent SEs that can be connected by information and control signals, $sw(v)$ is the number of control signals along the path v, and $l(v)$ is the total number of SEs along the path v.

As a rule, the better the modeling means being used, the more perfect the devices constructed with their help. In particular, striving to minimize the time complexity computed by formula (1), we will construct (in Sec. 2) a parallel optical multiplier whose size is considerably smaller and architecture is simpler than those of the circuit of multiplication of multidigit numbers from [1].

Before describing the structure of the multiplier, we will pay attention to one more distinctive feature of digital optical computations. In addition to active elements such as switches, logical circuits also contain passive elements that do not change their states as a result of definite physical processes. They include optical waveguides, waveguide bends, waveguide branchings, etc. In our case, it is especially important that optical connectors realizing the logical function OR are also passive elements. In particular, in [5], the *Y*-shaped connection of waveguides in a photon crystal (Fig. 4) is considered and it is shown that, when a defect of special form is created at the junction point of the waveguides, the optical energy is transmitted in the directions $a_1 - b$ or $a_2 - b$ rather than in the directions $a_1 - a_2$ or $a_2 - a_1$. Other passive devices for connection of waveguides, for example, a directional coupler [6, 7], are also widely used. Thus, the execution of the operation OR in optical calculators does not mean that a physical process runs that occupies some time; it is performed as a result of a special connection of channels along which optical signals are transmitted. This distinctive feature is used in Sec. 2, and the logical OR is denoted by a filled triangle and is not taken into account in determining the time complexity of circuits. It is worth noting that the multiplying circuit considered in Sec. 2 can also be realized without OR connectives if we use SEs with two information inputs and one output.

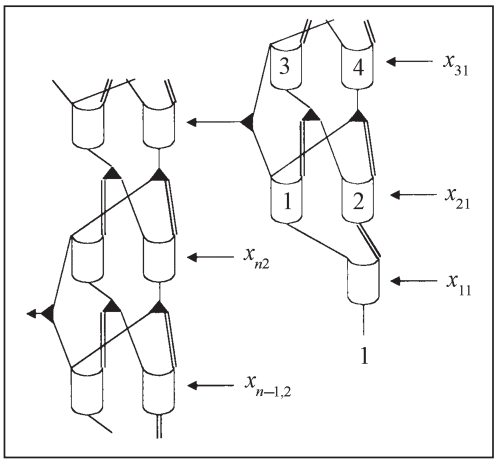

Fig. 5. Determination of the parity of the number of unities in a column and the carry to its adjacent column.

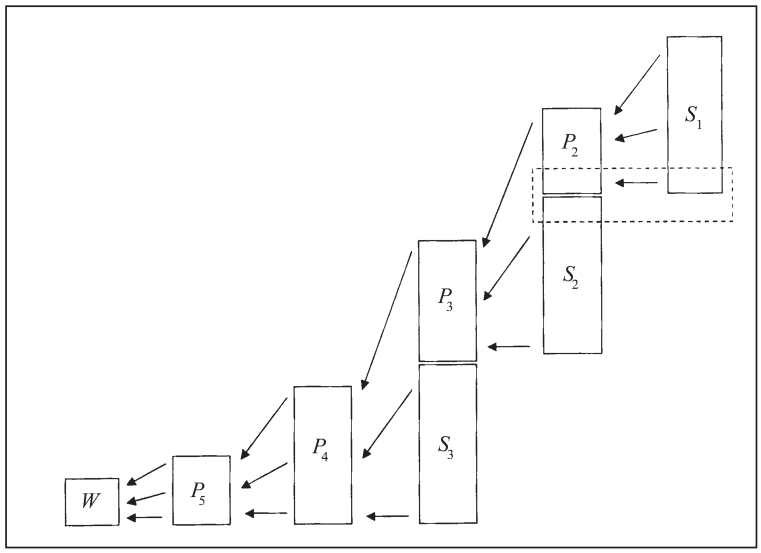

Fig. 6. Circuit for translation of multirow codes into one-row ones.

2. OPTICAL MULTIPLIER

The circuit that is described below and realizes the multiplication operation is the circuit proposed in [1] and improved in a definite sense. Both these circuits belong to the class of multilevel matrix multipliers. A matrix of partial products is applied to the input of the first level of this multiplier. At the output of the ith level, we obtain $k_i < k_{i-1}$ numbers whose sum is equal to the sought-for product. In particular, at the output of the last level, we obtain two numbers that should be added together.

The difference between multilevel matrix multipliers consist of the method of translating the sum of k_{i-1} numbers (some k_{i-1}-code) into the sum of k_i numbers (a k_i-row code). In particular, in [1], the matrix of addends is partitioned into cells and the circuit that is described in detail in [4] and is designed for the transformation of multirow codes into one-row ones is applied to each of these cells (hereafter, we call the multiplier from [1] cellular). An important advantage of the circuit from [4] is its two-clock switch time, but it also has two essential drawbacks. First, the circuit from [4] is rather long and if it is applied to a cell of size $a \times b$, then the length of the path over which an information signal is transmitted is proportional to the value of ab. To decrease the length of the circuit from [4] in a cellular multiplier, it is applied not to the entire matrix of partial products but to its fragments. Second, the size of the circuit from [4] leaves something to be desired since the result of summation of a b-bit numbers is proportional to the value of $a^2 b$. We will describe a circuit that makes it possible to perform a similar operation with the help of $O(ab)$ SEs. It is the parallel application of such circuits to fragments of the matrix of addends that forms the transformation performed at each or some levels of the multiplier described in this work.

2.1. A Circuit Translating a Multirow Code into a One-Row One. In constructing the circuit described below, as well as in designing the circuit from [4], we proceeded from the following fact: the ith bit of the sum is determined by the parity of the number of unities in the ith column of the matrix of addends with allowance made for all the carries arrived from the right bits. However, the method of determination of the parity of the number of unities in a column and the method of their transfer to the left columns are different in the mentioned circuits.

In the circuit from [4], for calculation of the number of unities, a one-cycle decoder translating an arbitrary binary vector into a vector of the form 0 . . . 01 . . . 1 with the same number of unities was used. Then the $2j$th output of the decoder was used as the jth information input of the decoder that processes the next column. Thus, a carry was realized, i.e., instead of each pair of unities, one unity was "written" in the adjacent column to the left. The decoder has quadratic space complexity with respect to the length of the vector being decoded and the bits of the vector are its control signals. Since decoders are connected only by information inputs/outputs, the entire circuit from [4] is switched during one clock cycle.

The parity of the number of unities in a vector can also be found with the help of a linear-size circuit, and we will replace the decoder by such a scheme. The same circuit will realize the transfer of pairs of carries to the adjacent column to the left. It may be noted that, in a linear circuit, information channels in which pairs of unities would be "accumulated" are

absent. They are also absent in a similar circuit that processes the column to the left and, hence, a unique method of introduction of carries into a circuit that processes the column to the left is their equating with some of its control inputs. The structure of a circuit that processes a column and also the technique of connection of circuits corresponding to adjacent columns are shown in Fig. 5.

Let us consider the principle of operation of the circuit presented in Fig. 5. Each pair of SEs located at one horizontal level is controlled by one signal. We note that the bits 01 or 10 arrive at the inputs of any pair of SEs controlled by the *ij*th signal x_{ij}, and the choice of a concrete pair of bits is determined by the parity of the number of unities in the vector $x_{1j},\ldots,x_{i-1,j}$. This can be easily proved by induction. Let us consider the SEs that are denoted by the numbers 1–4 in Fig. 5. We denote their inputs by $In_1 - In_4$, their 0-outputs by $Out_1^0 - Out_4^0$, and their 1-outputs by $Out_1^1 - Out_4^1$. If we have $x_{11}=0$, then we obtain $In_1=0$ and $In_2=1$, but if we have $x_{11}=1$, then, vice versa, we obtain $In_1=1$ and $In_2=0$. We now assume that SEs 1 and 2 form an arbitrary pair of SEs in the circuit that processes the first column. If we have $x_{21}=1$, $In_1=0$, and $In_2=1$, then we obtain $Out_1^0=Out_1^1=Out_2^0=0$, $Out_2^1=1$, $In_3=Out_1^0\vee Out_2^1=1$, and $In_4=Out_1^1\vee Out_2^0=0$. It is also obvious that we have $x_{21}=1$, $In_1=1$, $In_2=0\Rightarrow In_3=0$, $In_4=1$; $x_{21}=0$, $In_1=1$, $In_2=0\Rightarrow In_3=1$, $In_4=0$; $x_{21}=0$, $In_1=0$, $In_2=1\Rightarrow In_3=0$, and $In_4=1$. Thus, if we have $x_{ij}=0$, then the same signals arrive at the inputs of the $i+1$th pairs of SEs as at the inputs of the *i*th pair, and if we have $x_{ij}=1$, then the pair of signals 01 is replaced by 10 and vice versa. This implies that if the vector $x_{1j},\ldots,x_{i-1,j}$ contains an odd number of unities, then 10 arrives at the inputs of the pair of SEs controlled by the *ij*th signal x_{ij} and 01 arrives otherwise, i.e., the zero output of the last left element connected by a connective OR with the unit output of the last right element is equal to the oddness of the number of unities in the corresponding column of the matrix of addends.

Let us consider the process of transfer of a carry to the left column. Again, we assume that SEs 1 and 2 in Fig. 5 form an arbitrary pair of SEs that is controlled by the *ij*th signal x_{ij}. A carry signal must be generated in the case when x_{ij} forms the next pair of unities in the vector $x_{1j},\ldots,x_{ij}$, i.e., if we have $In_1=1$ and $x_{ij}=1$. In order that this signal influence the parity of the number of unities in the column to the left, it should be transformed into a control signal for the pair of SEs in the circuit that processes this column (see Fig. 5). It may be noted that carry signals cannot be generated by two adjacent bits of the vector $x_{1j},\ldots,x_{ij}$ and, hence, the carry generated by SE 1 is connected by a connective OR with the carry generated by SE 3. We will also pay attention to the fact that the carry signals from the right column are processed only after processing all the bits of the left column. Though, from a logical viewpoint, carry signals can be processed at any moment, for example, before the processing the bits of a column or alternately with such a processing, it is the final processing of them and precisely in the order in which they are generated by the right column makes it possible to optimize the time characteristics of the circuit, which will be shown below. The technique of connection of circuits processing several columns of a matrix of addends is represented in Fig. 6. We call the circuit obtained as a result of this connection *basic*.

Here, S_1, S_2, and S_3 are circuit fragments processing the bits of the matrix of addends, $P_2 - P_5$ are fragments processing carry bits, and *W* denotes a single-clock circuit computing the weight of the vector obtained at the output of the circuit P_5.

The structure of the circuit *W* is presented in Fig. 7. In this circuit, SEs form a decoder with a vector of the form $(v_k,\ldots,v_1)=(0\ldots010\ldots0)$ at its output. In this vector, unity is located at a position *w* if the outputs of the last circuit P_j contain *w* unities. If the outputs of the last circuit P_j contain no unity, than we obtain the zero vector at the output of the decoder. Given a vector $(v_k,\ldots,v_1)$, we can easily determine the number $w=w_m,\ldots,w_0$ with the help of a multibit OR circuits $w_0=v_1\vee v_3\vee v_5\vee\ldots$, $w_1=v_2\vee v_3\vee v_6\vee v_7\vee\ldots$, and $w_1=v_4\vee v_5\vee v_6\vee v_7\vee v_{12}\vee v_{13}\vee v_{14}\vee v_{15}\vee\ldots$. These circuits are realized with the help of two-digit OR circuits connected in the form of a treelike or linear structure and do not require any time for switching.

We denote by SP_i the entire circuit that processes the *i*th column and consists of fragments S_i and P_i. All these circuits are of the form that is presented in Fig. 5 and that corresponds to the rectangle enclosed by a dotted line in Fig. 6. Note that the processing of carries from the leftmost column, in addition to the fragments P_i located above the fragments S_i, requires some additional circuits P_j; their number does not exceed the number of fragments S_i and its exact value will be determined below. We will also show that fragments P_i at most double the size of the circuits that process columns. This means that the addition of *a b*-bit numbers requires $O(ab)$ SEs.

In contrast to the circuit from [4] that switches in two clock cycles, the switch time of the circuit being considered exceeds the number of columns, which is the payment for the decrease in the size, but the loss can be reduced to zero by a variation in lengths of columns. The idea consists of providing the condition under which an information signal does not wait for the corresponding control signal but, during the delay conditioned by switching, passes along the chain of SEs for which

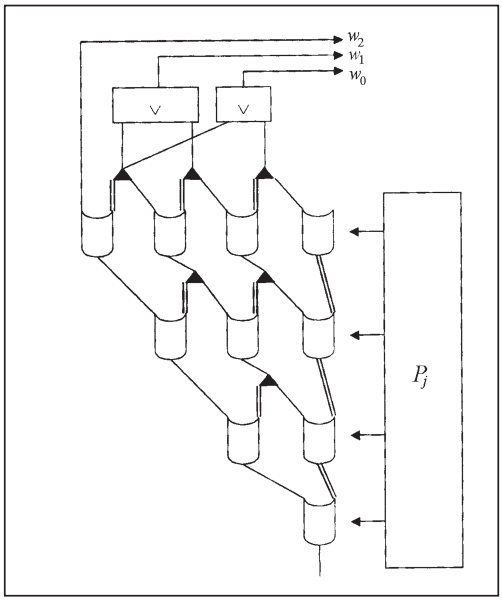

Fig. 7. The circuit W that processes high-order bits.

control signals are already computed. In other words, the lengths of circuits S_i should be selected so that they all simultaneously come into operation and, at the same time, the information signal should arrive to the last pair of SEs in a fragment P_i exactly at the moment of their switching.

Let L_{sw} be the number of SEs in circuits S_i or P_i through which the information signal passes during switching one SE. L_{sw} approximately equals t_{trans} / t_{sw} if t_{trans} and t_{sw} take into account the time of transformation of an information signal into a control one, time of passage of a signal along the branches of optical waveguides between switches, and other "overheads." LS_i, LP_i, and LSP_i denote the lengths of fragments S_i, P_i, and the entire circuit of processing the ith column, respectively, i.e., we have $LSP_i = LS_i + LP_i$.

Let us show that, to meet the above conditions of absence of "idle time" in the circuit, it is necessary and sufficient that the following equality be true when $i > 1$:

$$LS_i = (LS_1 + iL_{sw}) / 2. \tag{2}$$

By the construction of the basic circuit, the last SE in each column is switched after the time t_{sw} after the switching of the last SE in the previous column. During this time, the information signal IS in each circuit SP_i in which it has not yet reached its end has time to pass through L_{sw} SEs. Moreover, the IS in each column is propagated during the same time as the IS in the first column, i.e., any IS passes through LS_1 SEs even before the switching of the last SE in the first column. Thus, we have the relationship

$$LSP_i = LS_1 + (i-1)L_{sw}. \tag{3}$$

On the other hand, taking into account that to each two control signals in a circuit SP_{i-1} corresponds one control signal in the fragment P_i, we obtain the relationship

$$LP_i = LSP_{i-1} / 2. \tag{4}$$

Relationships (3) and (4) imply equality (2) since we have $LS_i = LSP_i - LP_i = LS_1 + (i-1)L_{sw} - LSP_{i-1} / 2 = LS_1 + (i-1)L_{sw} - (LS_1 + (i-2)L_{sw}) / 2 = (LS_1 + iL_{sw}) / 2$. In particular, as is obvious from relations (2)–(4), we have $LP_i < LS_i$ and, hence, the circuit size linearly depends on the number of bits in the matrix of addends.

Assuming that a circuit contains b columns S_i that process n bits in all and c columns P_j that do not belong to SP_i and that its rightmost column processes $a = LS_1$ bits, we determine the relationship between the parameters a, b, c, and n. The parameter b of the basic circuit is expressed through the parameter a and the circuit length n, namely, b is the largest integer that satisfies inequality (5) that follows from relationship (2),

$$a + \frac{1}{2}\sum_{i=2}^{b} (a + iL_{sw}) \le n. \tag{5}$$

The parameter a should be selected so that, for the integer b corresponding to it, relationship (5) is maximally close to equality. This makes it possible to avoid time losses connected with the delay of the beginning of propagation ISs through the circuit S_b if its length is too small. Among all the values of a that satisfy this condition, the smallest value should be chosen. This becomes obvious if we note that the operation time of the entire cascade of circuits SP_i equals $t_{trans}LSP_b + t_{sw}$, i.e., it decreases with decreasing LSP_b. To a smaller a corresponds a larger b and, hence, a smaller length of the circuit SP_b.

We will also investigate the following question: what is the expedient number of circuits P_j that do not belong to SP_j, i.e., estimate the value of the parameter c. These circuits cannot begin to simultaneously operate, and since the length of each next circuit P_j is half the length of the previous one, P_j must begin to operate at the moment that allows it to complete its operation after the time t_{sw} after the completion of operation of P_{j-1}. Therefore, the computation of all the bits of the sum with the help of the cascade of circuits P_j is inefficient from the viewpoint of time complexity. Instead, when the length of the vector of carries becomes sufficiently small, we will use one circuit W after the cascade consisting of c circuits P_j for computation of the remained bits of the sum. In determining the value of c, we proceed from the fact that the size of the circuit W should not "distort" the general estimate of space complexity. This size amounts to $(LP_c / 2)^2 / 2 = LP_c^2 / 8$ SEs. But if these LP_c bits will also be further processed by circuits P_j, then we will need $2(LP_c / 2 + LP_c / 4 + \ldots + 1) = 2LP_c - 2$ SEs. Thus, as soon as the inequality $LP_c^2 / 8 < 2LP_c - 2$ is fulfilled that is equivalent to inequality (6) since LP_c is an integer, the use of circuits $P_{c+1}, P_{c+2}, \ldots$ loses any meaning at all and their cascade should be replaced by one circuit W,

$$LP_c < 15. \tag{6}$$

The use of the circuit W in the case when we have $LP_c \geq 15$ will reduce the operation time of the basic circuit owing to the increase in its size.

We estimate the size and total operation time of the basic circuit. If we assume that the circuit W is absent and that all carries are processed by a cascade of circuits P_j, then a sufficiently exact upper bound of the circuit size can be easily obtained. To each of n inputs of the circuit corresponds two SEs. Moreover, each pair of bits generates its carry bit, each pair of carry bits generates one more such a bit, etc., i.e., no more than $(n / 2 + n / 4 + \ldots + 1) < n$ carry bits are generated in the aggregate and each of them is processed by two SEs. In total, we obtain that the circuit size is no more than $4n$ SEs. When $LP_c \geq 15$, this estimate can be larger by a factor of at most $LP_c^2 / 8 - 2LP_c + 2$. But if we have $LP_c < 15$, then the mentioned value will be negative and it also should be added to $4n$. Since we have $LP_c = LSP_b / 2^c = (a + (b-1)L_{sw}) / 2^c$, we obtain the following formula for the upper estimate of the basic circuit size:

$$4n + LP_c^2 / 8 - 2LP_c + 2, \text{ where } LP_c = (a + (b-1)L_{sw}) / 2^c. \tag{7}$$

The computation of the total operation time of the basic circuit is sufficiently easy, namely, all the SEs in the circuits $S_1 - S_b$ are first switched in parallel, then the IS is propagated through these circuits and, when it will attain the end of the circuit S_1, the last SEs of circuits $P_1 - P_{b+c}$ are sequentially switched without delays and the last SEs in each column of the circuit W are switched after them. The time of transfer of the information signal through the circuits $SP_2 - SP_b$ and $P_1 - P_c$ can be neglected since the IS is transferred during switchings, but one should take into account the time of transfer of the IS through the circuit W whose length is equal to $(a + (b-1)L_{sw}) / 2^{c+1}$. As a result, we obtain the following formula for the determination of the total operation time of the basic circuit:

$$(b + c + 1)t_{sw} + (a + (a + (b-1)L_{sw}) / 2^{c+1})t_{trans}. \tag{8}$$

Table 1 presents the characteristics of the basic circuit constructed from SEs realized with the help of Fabry–Perot microresonators ($t_{sw} = 8$ps, $t_{trans} = 0.033$ps, and $L_{sw} = 240$) for different values of n that were chosen so that the parameter a was equal to 0. In this case, the value of b was determined as the largest integer satisfying inequality (5) and, hence, the values of n presented in the table are the least n for given values of b. After fixing the values of a, b, and n, the parameter c varied from the largest value satisfying inequality (6) to a value that was less by 1–2 and thereby determined various ratios between time and space complexities. For circuits realized with the help of another element base, similar calculations can be performed.

2.2. Multiplying circuit. The matrices of addends that are located at each level of a multilevel matrix multiplier should be completely covered by the inputs of the circuits that translate multirow codes into one-row ones. Note that, though the input columns of each of such circuits can be arbitrarily formed from bits of the corresponding columns of the matrix of

TABLE 1

n	*a*, *b*, *c*	Size, SEs	Additional size, SEs	Operation time of the circuit, ps
120	0; 2; 4	480	0	56
	0; 2; 3	534	54	48
360	0; 3; 5	1440	0	72
	0; 3; 4	1494	54	64
	0; 3; 3	1772	332	57
720	0; 4; 6	2875	– 5	88
	0; 4; 5	2898	18	80
	0; 4; 4	3045	165	73
1200	0; 5; 6	4800	0	96
	0; 5; 5	4854	54	88
	0; 5; 4	5132	332	81
1800	0; 6; 7	7194	–6	112
	0; 6; 6	7206	6	104
	0; 6; 5	7299	99	97

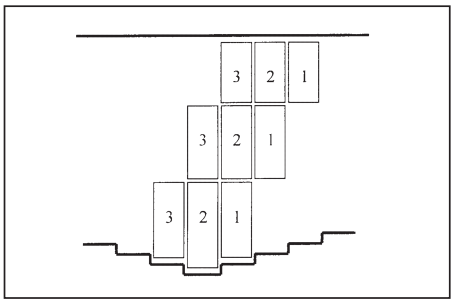

Fig. 8. Configuration of inputs of circuits that translate multirow codes into one-row ones.

addends, from the viewpoint of the decrease in the length of interelement connections, the arrangement of inputs in the form of a staircase such as that presented in Fig. 6 is most convenient. In this case, the entire matrix of partial products at the first level of the multiplier can be covered by the inputs of the circuits of translation of multirow codes as is shown in Fig. 8, where *i* denotes the inputs of the *i*th circuit. As a rule, this matrix is represented in the form of a parallelogram but it can also be represented in the form of a triangle by arranging the contacts in the left part of the parallelogram in a different way.

Since the size of the basic circuit linearly depends on the number of bits processed by it, a partitioning of the matrix of addends into horizontal strips each of which is processed by a collection of basic circuits will not decrease the total amount. Calculations have shown that such a partitioning does not essentially reduce the operation time of each basic circuit; therefore, we will assume that the inputs of these circuits cover the largest possible number of bits from the upper end of the matrix to its lower end. "Emptinesses" remain at the angles of the triangle of the matrix of addends, but they can be eliminated by a proper selection of the value for the parameter *a* of the corresponding basic circuits and also by replacing condition (2) by the inequality $LS_i < (LS_1 + iL_{sw})/2$. The unique requirement imposed on the circuits that cover bits at angles is that their operation time must not exceed the operation time of the largest basic circuit denoted by the number 2 in Fig. 8.

At the first level of the multiplier of *n*-bit numbers, the largest basic circuit will have *n* inputs. Its operation time is specified by formula (8) and is equal to the time of processing the first level. As is easily seen, the word length of the number into which *n* input bits are translated by the largest basic circuit is more than the word length of the number LSP_b by $b-1$, i.e., amounts to $b-1+\lfloor \log LSP_b \rfloor + 1 = b + \lfloor \log(a + (b-1)L_{sw}) \rfloor$ bits. As is obvious, this value can be the upper bound of the number of rows of the code into which an *n*-row matrix of partial products is translated by the circuit being considered. Computations showed that, for *n* within 10 thousand and for the values of $L_{sw} \geq 4$ (this inequality is fulfilled for all the considered types of element bases), this value does not exceed several tens and it is precisely the number of addends arriving at the second level of the multiplier. They can be processed with the help of the circuit considered in this work and with the help of the circuit from [1]. Against the background of the total space complexity, the size of the circuit from [1] applied to such a small amount of addends will not be considerably larger than the size of the circuit that is described in this work and realizes computations at the second level of the multiplier. However, the circuit from [1] can turn out to be faster and, hence, it makes sense to construct a combined multiplier that performs computations according to the scheme described in this work only at the first or at the first and second levels whose outputs become inputs of the circuit from [1].

Let us estimate the size of the circuit located at the first level of the multiplier. Since the lengths of circuits that translate a multirow code into a one-row one are different and the values of the parameters *a*, *b*, and *c* also vary, the use of formula (7) for estimation of the total size of the first level of the multiplier seems to be difficult. A sufficiently exact upper bound of this size can be obtained using the following approach similar to that used in estimating the size of the basic circuit: to each bit of the matrix of partial products corresponds two SEs, and the total number of such bits during multiplying two *n*-bit numbers will be equal to n^2, which yields $2n^2$ SEs. Moreover, each pair of bits generates its carry bit, each pair of carry bits generates one more carry bit, etc., i.e., the total number of produced bits will be no larger than $(n^2/2 + n^2/4 + \ldots + 1) < n^2$ carry bits, each bit being processed by two SEs. In total, we obtain that the size of the circuit is no larger than $4n^2$. We note that the size of the two-level multiplier from [1] is asymptotically larger since it is equal to

$O(n^{5/2})$ SEs, and the multiplier considered in this article in the case when it is constructed from Fabry–Perot microresonators and when the values of n amount to 10 thousand will also be two-level.

Let us make some comments on the value of the parameter c in basic circuits. As has been noted in item 2.1, it would make no sense to assign a value larger than the least number for which the inequality $LP_c < 15$ is fulfilled to this parameter. If this inequality is true, then the value of $4n^2$ remains the upper bound of the circuit size but, for circuits of larger length, i.e., located more closely to the center of the matrix of addends, it makes sense to decrease the value of this parameter since it is precisely these circuits that determine the time complexity of a level, and the operation time of the basic circuit decreases with decreasing the value of c. Therefore, designing basic circuits for concrete values of n and L_{sw} in the cases when this will essentially improve the time complexity of the largest basic circuit, we will chose c whose value is smaller by 1–3 than the least number satisfying the inequality $LP_c < 15$. In these cases, the time complexity of several basic circuits located at the center of the matrix of addends should be reduced to the time complexity of the largest basic circuit by selection of their parameters c. This approach leads to an excess of the space complexity $4n^2$ that should be taken into account. We assume that the largest basic circuit increases its size by v owing to an additional decrease in the parameter c. Then we consider that all the basic circuits in which the value of the parameter b is equal to the value of this parameter in the largest basic circuit also increase their sizes by v. In the circuits whose parameter b is smaller by unity, the parameter c can be closer by one to the value satisfying the inequality $LP_c < 15$ than the parameter c in the largest basic circuit. We assume that all such circuits also increase their sizes by the same value that, as well as the number of circuits, can be determined from Table 1 or from a similar table for another element base. For example, if we have $n = 400$ and circuits constructed form Fabry–Perot microresonators are considered, then, for the largest basic circuit, we have $a = 20$, $b = 3$, and the least value of c satisfying the inequality $LP_c < 15$ equals 5. If we put $c = 3$, then we have $LP_c = 62$, the excess of size of the largest basic circuit amounts to $LP_c^2 / 8 - 2LP_c + 2 = 359$ SEs, and, taking into account the contents of Table 1, the total excess of size can be computed as follows: $2 \cdot (400 - 360) \cdot 359 + 2 \cdot (360 - 120) \cdot 54 = 56640$ SEs. The described approach allows one to obtain the upper bound of the space complexity of the circuit.

3. COMPARATIVE ANALYSIS OF TIME AND SPACE COMPLEXITY OF MULTIPLIERS

Table 2 presents time and space characteristics of different circuits that translate multirow codes into two-row codes for SEs constructed from Fabry–Perot microresonators. Using the methods described above, the parameters a and b of all basic circuits are uniquely determined from the word length n of multipliers. The unique parameter that can be varied and that determines the parameters c of other basic circuits is the parameter c of the largest basic circuit. Let us select "reasonable" values of this parameter that make it possible to decrease the operation time of the largest basic circuit by 10–20% with increasing the circuit size by the same value. A further decrease in c will progressively decrease the operation time with increasing the circuit size.

When $n \le 1024$, after using the circuit described in this work at the first level, the number of addends at the second level will not exceed 12. The circuit from [1] is most efficient for addition of this number of $2n$-bit numbers and transforms the matrix of addends into a two-row code at one level, i.e., the entire circuit will be two-level.

In addition to the characteristics of the circuit considered in Sec. 2, for each n, we present the characteristics of a time-optimal circuit [1] and also an asymptotically fastest classical multilevel multiplier [8] constructed from the elements of the corresponding optical element base. At each level of this multiplier, a three-row code is translated into a two-row code. Switching circuits that realize this transformation are described in [1].

We note that, increasing the number of levels and operation time of the circuit from [1], one can reduce its size. With equating the time complexities of the circuit from [1] and the circuit proposed in this work, the latter will have a size advantage only about 20–30%, depending on n. However, a smaller number of levels of the multiplier is per se an essential advantage. Of importance is also the fact that the considered circuit of translation of multirow codes is flat and, taking into account subcircuits for processing carries, it can be placed on two planes parallel to the matrix of partial products (if $O(n)$ subcircuits of W are not taken into account). At the same time, a natural construction for the two-level multiplier from [1] consists of the placement of $O(n^{3/2})$ subcircuits from [4] perpendicularly to the matrix of addends.

It is also necessary to note that, in the case when optical-optical Mach–Zehnder switches in photon crystals are used in the capacity of the element base, the circuit described above has scarcely any size advantage over the multiplier from [1] and

TABLE 2

n	Circuit Described in This Paper		Cellular Multiplier [1]		Classical Multiplier Constructed from Optical Switches	
	Operation time, ps	Size, SEs	Operation time, ps	Size, SEs	Operation time, ps	Size, SEs
64	51	16400	29	133100	89	54600
128	58	84100	36	156700	97	219400
256	69	310600	40	918500	113	881300
512	77	1361000	43	4596000	130	3532000
1024	87	5051000	50	25436000	146	14139000

its operation time is less by 10–40% when the word length of multipliers is within 1000. This is explained by the smallness of the value of t_{sw} / t_{trans} for a Mach–Zehnder switch, this smallness implies a small cell size in the multiplier from [1], whereas the circuit considered above reduces the space complexity arising in this multiplier during processing larger cells.

REFERENCES

1. A. V. Anisimov and I. A. Zavadskyi, "Synchronous optical multipliers," Cybernetics and Systems Analysis, No. 4, 102–116 (2006).
2. C. Angulo Barrios, V. R. Almeida, R. R., Panepucci, et al., "Compact silicon tunable Fabry–Perot resonator with low power consumption," IEEE Photonics Technology Letters, **16**, No. 2, 506–508 (2004).
3. K. Asakawa, Y. Sugimoto, and Y. Watanabe, "Photonic crystal and quantum dot technologies for all-optical switch and logic device," New Journal of Physics, 8 (208) (2006), http://ej.iop.org/links/r9P_wqX0w/_BYBIl782xGbAYnnav5vpA/njp6_9_208.pdf.
4. I. Zavadskyi, "Multiplication using switching elements," Visnyk Kyivskogo Univ., Ser. Fiz.-Mat. Nauk, No. 4, 145–156 (1999).
5. "Topology optimization of asymmetric *Y*-junction for air-bride type photonic crystal slab waveguides," in: Proc. PECS-VII (2007), cmpweb.ameslab.gov/PECSVII/Abstracts/ WATANABEyoshinori.pdf.
6. F. Cuesta-Soto, A. Martinez, J. Garcia, et al., "All-optical switching structure based on a photonic crystal directional coupler," Optics Express, **12**, No. 1, 161–167 (2004).
7. Ts. Shyh-Lin (2) and Lu Chun-Yi, "BPM simulation and comparison of 1 x 2 directional waveguide coupling and *Y*-junction coupling silicon-on-insulator optical couplers," Fiber Integr. Opt., **21**, No. 6, 417–433 (2002).
8. M. A. Kartsev and V. A. Brik, Computing Systems and Synchronous Arithmetics [in Russian], Radio i Svyaz', Moscow (1981).